Remarks

Wentker discloses technologies related to "smart cards". Smart cards are, as Wentker discloses, large "credit card-sized plastic card[s] that include[] one or more semiconducter integrated circuits." These plastic cards provide an interface with a point-of-sale (POS) terminal, an ATM machine, or a vending machine, for example. Wentker, 4:43-63. These smart cards include a microprocessor, as well as both volatile memory (RAM) for use "as temporary storage for calculated results and as stack memory" (Wentker, 5:8-11), and non-volatile memory (EPROM or EEPROM) in which to store data such as card information and user information which must be retained when power is removed from the smart card, but which is also alterable over time. (Wentker, 5:13-21). Applications can also be loaded into the non-volatile memory of the smart cards. Wentker, 8:8-10.

These smart cards are quite different from the FPGAs discussed in claims 22-24, 48-50. An FPGA is a simple integrated circuit, on which resources such as memory are relatively scarce. An FPGA is typically configured using an off-chip bitstream which supplies configuration data, to indicate which of the electrical switches within the FPGA should be configured to create a desired circuit. A smart card, on the other hand, typically includes an 8 or 16 bit microcontroller with a cryptographic accelerator and a large amount of on-chip non-volatile memory, as discussed above. The smart card is programmed using application programs which are executed by a microprocessor on the smart card, much in the same manner as a conventional computer. Furthermore, the function of a smart card is quite different from that of an FPGA. A smart card's function is to implement security and store confidential data for a larger system, such as storing subscriber information for a cable TV set top box, or financial information for an ATM machine. The function of FPGAs, on the other hand, is to implement logic, not to securely store data. In 2001, when this application was first filed, FPGA manufacturers and customers were not concerned with bitstream security, and did not have a background in smart card systems. Thus it would not be obvious to those of skill in the art, as of the 2001 filing date of this application, to apply teachings from the non-analogous smart card technology to FPGAs.

For example, FPGAs do not have the non-volatile program storage memory that is taught by Wentker as being present in smart cards. FPGAs have volatile RAM memory to hold the bitstreams used to configure the FPGA while the FPGA is in operation, but they rely on external non-volatile memory to store the application bitstreams when the FPGA is not in operation. Therefore, with a smart card an application is loaded onto the smart card once, and it stays in the non-volatile memory inside the smart card indefinitely, whereas an FPGA bitstream is re-loaded into the FPGA's volatile memory from an external non-volatile memory every time that power is applied to the FPGA. This re-loading from external memory opens up a security vulnerability, by permitting the bitstream to be monitored as it is being input into the FPGA (Application, [0006]). Such monitoring is not possible in a smart card, because the smart card has the non-volatile memory containing the application embedded in the same plastic card as the microprocessor which executes the application. (Wentker, 4:64-67).

Wentker provides security for the applications by encrypting them before they are loaded onto the smart card. Wentker discloses that the data stored in the non-volatile memory can be encrypted, using encryption keys, a MAC creation key, and a key encryption key. Wentker, 7:20-25. The keys are all stored on the smart card, in the card manager 104 application. Wentker, 7:18-20. This application is stored on the smart card as well. Wentker does not disclose where the card manager 104 is stored on the smart card, but since card manager 104 is built on top of the run-time environment 102, as shown in FIG. 2, it is likely that card manager 104 is itself stored in the non-volatile memory of the smart card. Alternatively, card manager 104 may be stored in the ROM of the smart card. Since Wentker teaches storing its keys in the card manager 104, which is a software application stored in either the non-volatile memory or the ROM of the smart card, Wentker fails to disclose the claimed step of "storing a first secret key on an FPGA chip," as claimed in claims 22 and 48. As discussed above, a smart card is quite different from an FPGA chip. The closest that Wentker comes to disclosing an FPGA chip is in the boilerplate recitation of possible alternative embodiments of hardware devices that store an execute program code, where one of the list of devices is a "programmable logic device". However, Wentker provides no teachings of how such a PLD could be used in his smart card system, nor whether or how one could use such a PLD in the security scheme disclosed by Wentker; he merely recites it as a possible hardware circuit. At best, Wentker could be read to teach that the PLD could be a substitute for the microprocessor used in the smart card. Even if this were done, however, Wentker still teaches storing the keys in the card manager 104, which is software stored in the ROM or non-volatile memory. Wentker still does not teach storing a first secret key on an FPGA chip, as claimed in claims 22 and 48.

Furthermore, even if it is incorrectly assumed that the smart card discussed in Wentker is somehow an equivalent to the FPGA chip claimed in claims 22-24, 48-50, Wentker still fails to anticipate these claims. Claims 22-24, 48-50 recite " storing the message authentication code with bitstream information in a nonvolatile memory external to the FPGA chip." If the smart card is incorrectly equated to an FPGA chip, then the teachings of Wentker teach that the application program and the DAP is stored in a memory internal to the smart card, not external to it as required by claims 22-24, 48-50. Storing the bitstream information and message authentication code externally to the FPGA permits the use of superior density and operating speed SRAM FPGA technology (Application, [0004]), and provides an advantage over the teachings of Wentker, which teach the use of on-chip non-volatile memory (Wentker, 5:13-21).

Furthermore, the teachings of Wentker teach that the DAP is first created and appended to the application program by the application provider, without use of the smart card at all. Wentker, 14:66-15:51. The smart card does re-create the DAP, for purposes of verification that the installed program is in fact the same program that was initially created and assigned the DAP. Wentker's teachings are different from that claimed in claims 22-24, and are in fact the same as the prior art teachings discussed in the instant application at paragraphs [0040] – [0052], and distinguished from the teachings of claims 22-24, in which the MAC is calculated initially by the FPGA, not by an external system. As discussed in the instant application, the applicant's novel methods avoid the problems incurred by the methods taught by Wentker, wherein each smart card has to be supplied with a different version of the application, tailored to that smart card.

The Johnson reference (US 5,727,061) similarly fails to teach all of the limitations of claims 22-24, 48-50, even taken in combination with Wentker. First of all, Johnson teaches a communications system between a user access system (UAS), which is a portable computer, desktop, laptop or other similar computer, and a provider access system (PAS). Johnson, 7:11-15. Thus the teachings of Johnson are dealing with much larger scale components such as complex Intel microprocessors (Johnson, 7:7-8), which have processing capabilities far greater than those of a simple FPGA, and which has far larger memory capabilities than a simple FPGA. Johnson does not teach anything about FPGAs. In Johnson, the various information used for encryption/decryption purposes is stored on non-volatile memory 38, which is not part of any FPGA chip. Johnson, FIG. 2a, 7:4-11. Thus, Johnson also fails to disclose the claimed step of "storing a first secret key on an FPGA chip," as claimed in claims 22 and 48. Similarly, if the entire UAS 12 is incorrectly equated to a simple FPGA, then Johnson fails to teach the storing of any information, much less the required " message authentication code with bitstream information" in a memory external to the UAS 12.

Additionally, turning to claims 23 and 49, the Examiner is equating the claimed "copyright messages" with the "identification information" discussed in Wentker. However, in Wentker, the main concern is verifying that the application is authenticated by a supplier of the application, so that the application cannot tamper with the (presumably financial) data on the smartcard. The concern is not with piracy of the applications and enforcing copyright. The "identification information", as discussed in

Wentker, "identifies the entity that . . . provided the authentication pattern." Wentker, 21:32. In a smartcard environment, this entity is quite likely not to be the entity that creates and owns the copyright on the application software itself. For example, as discussed in Wentker, issuers such as banks, or regulatory agencies, are typically the entities providing the authentication pattern. Wentker, 20:54-21:6. These entites are not typically the copyright holders. Wentker provides no teaching that copyright messages are stored with the bitstream information, as claimed by claims 23 and 49.

Turning to claims 24 and 50, the DAP blocks discussed in Wentker do not identify the "identity of the customer to whom the pirated FPGA was originally supplied", as required by claims 24 and 50. Instead, the DAP blocks identify the entity that provided the authentication pattern, as discussed above. These entities are issuers or regulatory agencies, not customers. The DAP blocks used in Wentker are not intended for use as an anti-piracy measure, to identify the intended or original customers of pirated FPGA designs. The DAP blocks are intended to assure a particular smart card that the program it is receiving is in fact the same program as the one which was shipped (i.e. that it was not modified in transit), and that the program has been authenticated by some kind of authentication entity.

Conclusion

Prompt and favorable action on the merits of the claims is earnestly solicited. Should the Examiner have any questions or comments, the undersigned can be reached at (949) 567-6700.

The Commissioner is authorized to charge any fee which may be required in connection with this Amendment to deposit account No. 15-0665.

Respectfully submitted,
ORRICK, HERRINGTON & SUTCLIFFE LLP

Dated: January 4, 2006

By: Donald Daybell
Donald Daybell
Reg. No. 50,877

Orrick, Herrington & Sutcliffe LLP
4 Park Plaza, Suite 1600
Irvine, CA 92614-2558
Tel. 949-567-6700
Fax: 949-567-6710